

EJERCICIO 2

+ public class Auto {
 private int numeroRegistro;
 private String modelo;
 public double Precio { get; set; }
 public Auto(int registro, string modelo) {
 this.numeroRegistro = registro;
 this.modelo = modelo;
 }
 public int NroRegistro() {
 return this.numeroRegistro;
 }
 public override string ToString() {
 return \$"{{this.numeroRegistro}}; {{this.modelo}}";
 }
}

```

+ public class Camion {
    private DateTime fecha;
    private int cantidad;
    public int NroRegistro { get; set; }
    public Camion(DateTime fecha, int cantidad) {
        this.fecha = fecha;
        this.cantidad = cantidad;
    }
    private Stack<Auto> transporte = new Stack<Auto>();
    public void CargarVehiculo(Auto unAuto) {
        if (transporte.Count < cantidad) {
            transporte.Push(unAuto);
        }
    }
    public Auto RetirarVehiculo() {
        Auto v = null;
        if (transporte.Count > 0) {
            v = transporte.Pop();
        }
        return v;
    }
    public string[] VerCarga() {
        string[] Carga = new string[transporte.Count];
        int i = 0;
        foreach (Vehiculo v in transporte.Count) {
            Carga[i] = v.ToString();
        }
        return Carga;
    }
    public double ValorAsegurado() {
        return valorAsegurado;
    }
    public int CantidadVehiculos() {
        return transporte.Count;
    }
    public override string ToString() {
        return $"{this.NroRegistro} - {this.fecha.ToString("dd-MM-yyyy")}";
    }
}

public double ValorAsegurado {
    get {
        double valorAsegurado = 0;
        foreach (Auto auto in transporte) {
            valorAsegurado += auto.Precio;
        }
        return valorAsegurado;
    }
}

```

+ public class Servicio {

private List<Auto> listaAutos = new List<Auto>();
private List<Camion> listaCamiones = new List<Camion>();
public int NroOrden { get; set; }

public int GenerarCamion (DateTime fecha, int capacidad) {

Camion camion = new Camion (fecha, capacidad);
listaCamiones.Add (camion);
NroOrden++;
camion.NroRegistro = NroOrden;

} return camion.NroRegistro;

public void RecibirCamion (Camion unCamion) {

listaCamiones.Add (unCamion);

public Auto DescargarCamion (int nroOrden) {

Camion camionBuscado = BuscarCamion (nroOrden);

Auto v = null;

if (camionBuscado != null) {

v = camionBuscado.RetirarVehiculo();

listaAutos.Add (v);

return v;

private Camion BuscarCamion (int nro) {

int i = 0;

int pos = -1;

while (pos == -1 && i < listaCamiones.Count) {

if (listaCamiones [i].NroRegistro == nro) {

pos = i;

i++;

} return v;

Camion c = null;

```
    if (pos != -1) {
        C = listaCamiones[pos];
    }
    return C;
}
```

```
public void CargarCamion (int nroOrden, Auto unAuto) {
    Camion camionBuscado = BuscarCamion (nroOrden);
    if (camionBuscado != null) {
        camionBuscado.CargarVehiculo(unAuto);
        listaAutos.Remove(unAuto);
    }
}
```

```
public void CerrarCamion (int nro) {
    Camion camion = BuscarCamion (nro);
    if (camion != null) {
        string path = camion.ToString() + ".csv";
        FileStream fs = null;
        StreamWriter sw = null;
        try {
            fs = new FileStream (path, FileMode.Create, FileAccess.Write);
            sw = new StreamWriter (fs);
            sw.WriteLine ("nroRegistro;modelo");
            foreach (string linea in camion.VerCarga ()) {
                sw.WriteLine (linea);
            }
        } catch (Exception ex) { throw new ex; }
        finally {
            if (sw != null) sw.Close();
            if (fs != null) fs.Close();
        }
    }
}
```

```

④ public class FormPrincipal {
    Servicio servicio = new Servicio();
    int num;

    private void btnCrearCamion_Click() {
        DateTime fecha = dtpFecha.Value;
        int capacidad = Convert.ToInt32(tbCapacidad.Text);
        int num = servicio.GenerarCamion(fecha, capacidad);
        MessageBox.Show("Camion Creado");
        #region Limpio campos
        tbFecha.Clear();
        dtp.Value = DateTime.Now;
        #endregion
    }

    private void btnCargar_Click() {
        Auto VehiculoSelected = lsbAutos.SelectedItem as Auto;
        if (VehiculoSelected != null) {
            servicio.CargarCamion(num, VehiculoSelected);
            lsbAutos.Items.Remove(VehiculoSelected);
            MessageBox.Show("Vehiculo cargado");
        } else {
            MessageBox.Show("Debes seleccionar un Vehiculo de la lista para Cargarlo");
        }
    }

    private void btnDescargar_Click() {
        Auto Vehiculo = servicio.DescargarCamion(num);
        lsbAutos.Items.Add(Vehiculo);
    }
}

```

```
private void btnCerrarCamion_Click(object sender, EventArgs e)
```

```
{
```

```
    servicio.CerrarCamion(num);
```

```
}
```

```
private void btnRecibirCamion_Click(object sender, EventArgs e)
```

```
{
```

```
    openFileDialog.Filter = "(csv) | *.csv | (txt) | *.txt";
```

```
    openFileDialog.Title = "Importacion de Vehiculos";
```

```
    if (openFileDialog.ShowDialog() == DialogResult.OK)
```

```
    {
        FileStream fs = null;
```

```
        StreamReader sr = null;
```

```
        string path = openFileDialog.FileName;
```

```
        try
```

```
        {
            foreach (Auto auto in listaCamiones)
```

```
                fs = new FileStream(path, FileMode.Open, FileAccess.Read);
```

```
                sr = new StreamReader(fs);
```

```
                sr.ReadLine();
```

```
                while (!sr.EndOfStream)
```

```
                {
                    string linea = sr.ReadLine();
```

```
                    if (linea != null)
```

```
                        string[] splitResult = linea.Split(';');
```

```
                        int nroRegistro = Convert.ToInt32(splitResult[0].Trim());
```

```
                        string modelo = splitResult[1].Trim();
```

```
                        double precio = Convert.ToDouble(splitResult[2].Trim()));
```

```
                        Auto auto = new Auto(nroRegistro, modelo);
```

```
                        auto.Precio = precio;
```

```
                        lstRecibidos.Items.Add(auto);
```

```
                    }
```

```
                    Camion camionRecibido = new Camion(DateTime.Now, lstRecibidos.Items.Count);
```

```
                    sistema.RecibirCamion(camionRecibido);
```

```
                    foreach (Auto auto in lstRecibidos.Items)
```

```
                    {
                        servicio.CargarCamion(num, auto);
```

```
                }
```

```
                catch (Exception ex)
```

```
                {
                    MessageBox.Show(ex.Message, "ERROR EN LA IMPORTACION",
```

```
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
```

```
            }
```

```
            finally
```

```
            {
                if (sr != null) sr.Close();
```

```
                if (fs != null) fs.Close();
```

```

private void bthDescargarCamion_Click(...) {
    if (num > 0) {
        Auto auto = servicio.DescargarCamion(num);
        while (auto != null) {
            lsbAutos.Items.Add(auto);
            lsbRetirados.Items.Remove(auto);
            auto = servicio.DescargarCamion(num);
        }
    }
}

```

```

private void FormPrincipal_Load(...) {
    ④ Deserializar
    List<Auto> autos = new List<Autos> {
        new Auto(100, "Audi AA"),
        new Auto(101, "Audi BB"),
        new Auto(102, "Corsa BZ"),
        new Auto(103, "Corsa Foli")
    };
}

```

Camion camion = new Camion(DateTime.Now, 0);
 camion.NroRegistro = sistema.NroOrden;
 servicio.RecibirCamion(camion);

```

foreach (Auto auto in autos) {
    camion
    .NroRegistro
    servicio.CargarCamion(" ", auto);
}

```

lsbAutos.Items.AddRange (autos.ToAuto);

④ FileStream fs = null;

try {

```

    fs = new FileStream("datos.bin", FileMode.Open, FileAccess.Read);
    BinaryFormatter bf = new BinaryFormatter();
    servicio = bf.Deserialize(fs) as Sistema;
}

```

}

catch (Exception ex) { MessageBox.Show(ex.Message, "ERROR AL DESERIALIZAR", MessageBoxButtons.OK, MessageBoxIcon.Error); }

finally {

if (fs != null) fs.Close();

Funcionamiento
Lógico

```
private void FormPrincipal_Closing( ) {
```

```
    FileStream fs = null;
```

```
    try {
```

```
        fs = new FileStream("datos.bin", FileMode.Create, FileAccess.Write);  
        BinaryFormatter bf = new BinaryFormatter();  
        bf.Serialize(fs, servicio);
```

```
    } catch (Exception ex) {
```

```
        MessageBox.Show("ex.Message", "ERROR AL SERIALIZAR", MessageBoxButtons.OK,  
        MessageBoxIcon.Error);
```

```
    } finally {
```

```
        if (fs != null) fs.Close();
```

```
}
```

```
}
```